

# An Algorithmic Framework for Computing Validation Performance Bounds by Using Suboptimal Models

Yoshiki Suzuki

Department of Engineering  
Nagoya Institute of Technology  
Nagoya, Japan  
suzuki.mllab.nit@gmail.com

Kohei Ogawa

Department of Engineering  
Nagoya Institute of Technology  
Nagoya, Japan  
ogawa.mllab.nit@gmail.com

Yuki Shinmura

Department of Engineering  
Nagoya Institute of Technology  
Nagoya, Japan  
shinmura.mllab.nit@gmail.com

Ichiro Takeuchi\*

Department of Engineering  
Nagoya Institute of Technology  
Nagoya, Japan  
takeuchi.ichiro@nitech.ac.jp

February 10, 2014

---

\*Corresponding author

## Abstract

Practical model building processes are often time-consuming because many different models must be trained and validated. In this paper, we introduce a novel algorithm that can be used for computing the lower and the upper bounds of model validation errors without actually training the model itself. A key idea behind our algorithm is using a side information available from a *suboptimal model*. If a reasonably good suboptimal model is available, our algorithm can compute lower and upper bounds of many useful quantities for making inferences on the unknown target model. We demonstrate the advantage of our algorithm in the context of model selection for regularized learning problems.

**Keywords:** model selection, approximate regularization path, convex optimization

# 1 Introduction

In practical model building processes, it is often required to train a large number of multiple different models. Those models are usually evaluated based on a generalization performance measure such as the validation error (e.g., mis-classification error rate on a validation data set). When the training algorithm of each of those models is formulated as an optimization problem, the entire model building process would be quite time-consuming. It is, however, important to note that the final goal of model building is to find the single best model. It means that we only need the validation error for the rest of the models and the model itself is not necessary. If we could compute the validation error of a model without actually training it, model building processes would be much more efficient.

In this paper, we introduce a novel algorithm for a class of regularized learning problems. Our algorithm can be used for computing the lower and the upper bounds of the validation error without actually solving the training optimization problem. Instead of computing the validation error directly from the trained model itself, our algorithm uses a side information available from a *suboptimal* model. If we have a reasonably good suboptimal model that is sufficiently close to the target model, our algorithm can provide the bounds of the validation error.

Our algorithm is especially useful in model selection for regularized learning problems, where a sequence of models with various regularization parameters are trained and validated. In this scenario, an already trained model with a certain regularization parameter can be used as the suboptimal model for our algorithm. Then, we can compute the validation error bounds of other unknown models associated with other regularization parameters. If the validation error lower bound of a model is larger than the smallest value obtained so far, we can skip training that model.

The basic idea behind our algorithm is computing a closed convex domain in the solution space in which we only know that the optimal solution exists, but the optimal solution itself is unknown. If such a closed convex domain is available, it is often possible to compute the bounds of a quantity depending on the unknown optimal solution. For a certain class of regularized learning problems, we show that such a domain can be easily derived and the bounds can be analytically computed based on a side information available from a suboptimal model. This algorithmic trick is inspired from a recent study on *safe screening* in the context of sparse modeling [5].

Our algorithm has a connection with recent studies on *approximate regularization path* [11, 7, 8]. Its key property is the ability to compute the lower bounds of the objective values of the training optimization problems. This property is useful for computing a regularization path with  $\varepsilon$ -approximation guarantee. In this context, our algorithm can be considered as a variant of such approximate regularization path algorithms. Instead of bounding the objective values, our algorithm can compute an  $\varepsilon$ -approximate regularization path in terms of validation errors, which is more useful for model selection purpose.

Our main contribution in this paper is to implement the above idea in a general algorithmic framework, and show that it can be useful in many practical machine learning tasks. Although we mainly focus on model

selection for binary classification problems, our algorithm can be applied to any learning problems defined with a convex loss function and an  $L_2$  regularizer. It can compute the lower and the upper bounds of many useful quantities for making inferences on unknown target models. To the best of our knowledge, there are no other previously known algorithms that can compute practically useful bounds for various types of model evaluation performances.

## 2 Problem Setup and Basic Idea

**Notations** For any natural number  $n$ , we define  $[n] := \{1, \dots, n\}$ . A real  $n$ -vector is denoted as  $v \in \mathbb{R}^n$  and  $v^\top$  indicates the transpose of the vector. Unless otherwise stated,  $\|\cdot\|$  is a Euclidean norm.

**Problem setup** Let us denote the training set as  $\{(x_i, y_i)\}_{i \in [n]}$ , where  $x_i \in \mathcal{X}$  is the input vector in the input space  $\mathcal{X}$  and  $y_i \in \{\pm 1\}$  is the binary class label. Let  $\phi : \mathcal{X} \rightarrow \mathcal{F}$  be a feature map associated with a kernel  $K$ . We consider a linear model in the feature space  $\mathcal{F}$  in the following form:

$$f(x) = \phi(x)^\top w,$$

where  $w \in \mathcal{F}$  is the vector of coefficients. For simplicity, we denote  $\phi_i := \phi(x_i), i \in [n]$ . We consider the following class of  $L_2$  regularized convex learning problems:

$$w_C^* := \arg \min_{w \in \mathcal{F}} \frac{1}{2} \|w\|^2 + C \sum_{i \in [n]} \ell(y_i, \phi_i^\top w), \quad (1)$$

where  $\frac{1}{2} \|w\|^2$  is an  $L_2$  regularization term,  $\ell$  is a convex loss function, and  $C > 0$  is the regularization parameter for controlling the balance between the two terms. We denote the optimal solution as  $w_C^*$  in order to clarify that it is the optimal solution associated with the regularization parameter  $C$ . With a slight abuse of notation, we use the following simplified notation when there is no ambiguity:

$$\ell_i(w) := \ell(y_i, \phi_i^\top w).$$

**Basic idea** In this paper, we develop a general algorithmic framework for computing the lower and the upper bounds of the inner product  $\theta^\top w_C^*$  for an arbitrary vector  $\theta \in \mathcal{F}$  *without actually solving the optimization problem* for  $w_C^*$ . We denote the lower and the upper bounds as  $b_{lo}(\theta^\top w_C^*)$  and  $b_{up}(\theta^\top w_C^*)$ , respectively, i.e.,

$$b_{lo}(\theta^\top w_C^*) \leq \theta^\top w_C^* \leq b_{up}(\theta^\top w_C^*).$$

We will demonstrate that this framework is quite useful in many practical machine learning tasks.

If we have a validation data set  $\{(x'_i, y'_i)\}_{i \in [n']}$  for a binary classification problem with  $x'_i \in \mathcal{X}$  and  $y'_i \in \{\pm 1\}$ , the mis-classification error rate

$$\frac{1}{n'} \sum_{i \in [n']} I \{y'_i \neq \text{sgn}(\phi_i'^\top w_C^*)\}$$

can be bounded from below and above by

$$\frac{1}{n'} \left( \sum_{i:y'_i=+1} I\{b_{up}(\phi'_i{}^\top w_C^*) < 0\} + \sum_{i:y'_i=-1} I\{b_{lo}(\phi'_i{}^\top w_C^*) > 0\} \right), \quad (2)$$

and

$$1 - \frac{1}{n'} \left( \sum_{i:y'_i=+1} I\{b_{lo}(\phi'_i{}^\top w_C^*) > 0\} + \sum_{i:y'_i=-1} I\{b_{up}(\phi'_i{}^\top w_C^*) < 0\} \right), \quad (3)$$

respectively, where  $I(\cdot)$  is the indicator function,  $\text{sgn}(\cdot)$  is the sign function, and  $\phi'_i := \phi(x'_i)$ .

Although we focus in this paper on the problem of computing validation error bounds for binary classification problems, our framework for bounding  $\theta^\top w_C^*$  is far more general. It can be used for computing the lower and the upper bounds of many useful quantities for validation, inference and prediction on various models.

Our basic algorithmic idea for computing the bounds of  $\theta^\top w_C^*$  is as follows. Suppose that we only know that the optimal solution  $w_C^*$  is somewhere in a closed convex domain  $\mathcal{S} \in \mathcal{F}$ , but we do not know the optimal solution  $w_C^*$  itself. In such a case, the lower and the upper bounds of  $\theta^\top w_C^*$  can be obtained by solving the following minimization and maximization problems:

$$b_{lo}(\theta^\top w_C^*) := \min_{w \in \mathcal{S}} \theta^\top w, \quad (4a)$$

$$b_{up}(\theta^\top w_C^*) := \max_{w \in \mathcal{S}} \theta^\top w. \quad (4b)$$

We later show that, for the class of regularized learning problems in (1), we can easily find such a closed convex domain  $\mathcal{S}$ , and the lower and the upper bounds in the forms of (4) can be analytically computed.

This algorithmic trick is inspired from a recent study on *safe screening* in the context of sparse modeling [5]. Safe screening enables to identify and screen out a part of the sparse model coefficients which turn out to be 0 at the optimal solution *before* actually training the model. Although our problem setup and goal are totally different, some of the algorithmic and proof techniques developed in [5] and the subsequent studies [22, 21, 3, 15, 13, 18, 19, 20, 16, 17, 12] are useful for our algorithm development (see Appendix 5 for more discussion on the relation between our approach and safe screening).

### 3 Bounds by Suboptimal Models

In this section we present our main results. In Theorem 1, we first describe our general result for computing the lower and the upper bounds of a quantity depending on the unknown optimal solution. In Theorem 2, we focus on model selection scenario for regularized learning problems, where we derive the lower and the upper bounds represented as the functions of the regularization parameter  $C$ .

**Theorem 1.** *Let  $\tilde{w} \in \mathcal{F}$  be an arbitrary vector in the feature space. Then,*

$$w_C^* \in \mathcal{S} := \{w \mid \|w - \tilde{w}\| \leq r\}, \quad (5)$$

i.e., the optimal solution  $w_C^*$  is in the ball  $\mathcal{S}$  whose center  $m \in \mathcal{F}$  and the radius  $r > 0$  are defined as

$$m := \frac{1}{2} \left( \tilde{w} - C \sum_{i \in [n]} \nabla \ell_i(\tilde{w}) \right), \quad (6a)$$

$$r := \frac{1}{2} \left\| \tilde{w} + C \sum_{i \in [n]} \nabla \ell_i(\tilde{w}) \right\|, \quad (6b)$$

where  $\nabla \ell_i(\tilde{w}) \in \mathcal{F}$  is the gradient vector of  $\ell_i$  at  $w = \tilde{w}$  when  $\ell_i$  is differentiable at  $\tilde{w}$ , while it is an arbitrary subgradient vector of  $\ell_i$  at  $w = \tilde{w}$  when  $\ell_i$  is non-differentiable at  $\tilde{w}$ .

It indicates that, for any  $\theta \in \mathcal{F}$ , the inner product  $\theta^\top w_C^*$  are bounded as

$$\theta^\top m - \|\theta\|r \leq \theta^\top w_C^* \leq \theta^\top m + \|\theta\|r,$$

i.e., the lower and the upper bounds are written as

$$b_{lo}(\theta^\top w_C^*) := \theta^\top m - \|\theta\|r, \quad (7a)$$

$$b_{up}(\theta^\top w_C^*) := \theta^\top m + \|\theta\|r. \quad (7b)$$

The proof of Theorem 1 is presented in Appendix A.

Theorem 1 is quite general because an arbitrary *suboptimal solution*  $\tilde{w} \in \mathcal{F}$  can be used for computing the bounds. However, it is important to note that, if we do not have a reasonably *good* suboptimal solution, the bounds in (7) could be quite loose and practically useless. We could roughly say that the bounds are tight when the suboptimal solution  $\tilde{w}$  is close to the optimal solution  $w_C^*$  (see § 5 for simple simulation results on this issue). The tightness of the bounds also depends on the curvature of the objective function<sup>1</sup>.

The following special case is very useful in the context of model selection for regularized learning problems. If we regard the optimal solution with a different regularization parameter  $\tilde{C} > 0$  as the suboptimal solution in Theorem 1, i.e., if we set  $\tilde{w} := w_{\tilde{C}}^*$  for a certain  $\tilde{C} > 0$ , the lower and the upper bounds of  $\theta^\top w_C^*$  are represented in simple interpretable forms.

**Theorem 2.** Let  $w_C^*$  be the optimal solution of the problem (1) for a regularization parameter  $\tilde{C} > 0$ . Then, for any  $\theta \in \mathcal{F}$ , the lower and the upper bounds of the inner product  $\theta^\top w_C^*$  are written as

$$b_{lo}(\theta^\top w_C^*) = \begin{cases} \frac{1}{2}(\theta^\top w_{\tilde{C}}^* + \|\theta\|\|w_{\tilde{C}}^*\|) + \frac{C}{2\tilde{C}}(\theta^\top w_{\tilde{C}}^* - \|\theta\|\|w_{\tilde{C}}^*\|) & \text{if } \tilde{C} < C, \\ \frac{1}{2}(\theta^\top w_{\tilde{C}}^* - \|\theta\|\|w_{\tilde{C}}^*\|) + \frac{C}{2\tilde{C}}(\theta^\top w_{\tilde{C}}^* + \|\theta\|\|w_{\tilde{C}}^*\|) & \text{if } \tilde{C} > C, \end{cases} \quad (8a)$$

$$b_{up}(\theta^\top w_C^*) = \begin{cases} \frac{1}{2}(\theta^\top w_{\tilde{C}}^* - \|\theta\|\|w_{\tilde{C}}^*\|) + \frac{C}{2\tilde{C}}(\theta^\top w_{\tilde{C}}^* + \|\theta\|\|w_{\tilde{C}}^*\|) & \text{if } \tilde{C} < C, \\ \frac{1}{2}(\theta^\top w_{\tilde{C}}^* + \|\theta\|\|w_{\tilde{C}}^*\|) + \frac{C}{2\tilde{C}}(\theta^\top w_{\tilde{C}}^* - \|\theta\|\|w_{\tilde{C}}^*\|) & \text{if } \tilde{C} > C. \end{cases} \quad (8b)$$

The proof of Theorem 2 is presented in Appendix A.

Interestingly, the bounds in (8) are represented as the functions of the regularization parameter  $C$ . It implies that, once we compute the optimal solution associated with a regularization parameter  $\tilde{C}$ , we can

---

<sup>1</sup> Since Theorem 1 tells that the solution is in a ball, the tightness of the bounds are closely related to the radius  $r$ . When the loss function  $\ell_i$  is differentiable and the optimal solution  $w_C^*$  itself is used as the suboptimal model  $\tilde{w}$  in Theorem 1, we could see that the radius is 0, i.e.,  $r = \frac{1}{2}\|w_C^* + C \sum_{i \in [n]} \nabla \ell_i(w_C^*)\| = 0$ , and the bounds in (7) are exact.

obtain a continuum path of the lower and the upper bounds of  $\theta^\top w_C^*$  parametrized by the regularization parameter  $C$ . The following corollary describes a few important properties of these parametrized bounds.

**Corollary 3.** (i) The lower bound (8a) is monotonically decreasing with  $C$  for  $C > \tilde{C}$ , and monotonically increasing with  $C$  for  $C < \tilde{C}$ . Similarly, the upper bound (8b) is monotonically increasing with  $C$  for  $C > \tilde{C}$ , and monotonically decreasing with  $C$  for  $C < \tilde{C}$ . (ii) Furthermore, the lower and the upper bounds converge to  $\theta^\top w_{\tilde{C}}^*$  as  $C$  approaches to  $\tilde{C}$ .

*Proof.* The part (i) can be easily proved by noting that

$$\theta^\top w_{\tilde{C}}^* - \|\theta\| \|w_{\tilde{C}}^*\| \leq 0 \text{ and } \theta^\top w_{\tilde{C}}^* + \|\theta\| \|w_{\tilde{C}}^*\| \geq 0$$

from the Cauchy-Schwartz inequality. For the part (ii), it is also clear to note that

$$\lim_{C \rightarrow \tilde{C}} b_{lo}(\theta^\top w_C^*) = \lim_{C \rightarrow \tilde{C}} b_{up}(\theta^\top w_C^*) = \theta^\top w_{\tilde{C}}^*.$$

□

**Bounds in the Intersection of Two Balls** If we have two suboptimal models  $\tilde{w}_1, \tilde{w}_2 \in \mathcal{F}$ , the optimal solution  $w_C^*$  is in the intersection of the two corresponding balls  $\mathcal{S}_1$  and  $\mathcal{S}_2$ . Since the intersection is smaller than each ball by definition, the bounds  $\min_{w \in \mathcal{S}_1 \cap \mathcal{S}_2} \theta^\top w$  and  $\max_{w \in \mathcal{S}_1 \cap \mathcal{S}_2} \theta^\top w$  are tighter than those obtained from a single ball. If the two balls are denoted as  $\mathcal{S}_1 := \{w \mid \|w - m_1\| \leq r_1\}$  and  $\mathcal{S}_2 := \{w \mid \|w - m_2\| \leq r_2\}$ , using the Lagrange multiplier methods (and tedious algebraic computation), the lower and the upper bounds in the intersection are computed as follows:

$$\min_{w \in \mathcal{S}_1 \cap \mathcal{S}_2} \theta^\top w = \begin{cases} \min_{w \in \mathcal{S}_1} \theta^\top w, & \text{if } \frac{-\theta^\top \alpha}{\|\theta\| \|\alpha\|} < \frac{\beta - \|\alpha\|}{r_1}, \\ \min_{w \in \mathcal{S}_2} \theta^\top w, & \text{if } \frac{\beta}{r_2} < \frac{-\theta^\top \alpha}{\|\theta\| \|\alpha\|}, \\ \theta^\top \gamma - \delta \left( \|\theta\|^2 - \frac{\|\theta^\top \alpha\|^2}{\|\alpha\|^2} \right)^{\frac{1}{2}}, & \text{otherwise.} \end{cases}$$

$$\max_{w \in \mathcal{S}_1 \cap \mathcal{S}_2} \theta^\top w = \begin{cases} \max_{w \in \mathcal{S}_1} \theta^\top w, & \text{if } \frac{\theta^\top \alpha}{\|\theta\| \|\alpha\|} < \frac{\beta - \|\alpha\|}{r_1}, \\ \max_{w \in \mathcal{S}_2} \theta^\top w, & \text{if } \frac{\beta}{r_2} < \frac{\theta^\top \alpha}{\|\theta\| \|\alpha\|}, \\ \theta^\top \gamma + \delta \left( \|\theta\|^2 - \frac{\|\theta^\top \alpha\|^2}{\|\alpha\|^2} \right)^{\frac{1}{2}}, & \text{otherwise,} \end{cases}$$

where  $\alpha := m_1 - m_2$ ,  $\beta := (\|\alpha\|^2 + r_2^2 - r_1^2)/(2\alpha)$ ,  $\gamma := m_2 + \beta\alpha/\|\alpha\|$ ,  $\delta := (r_2^2 - \beta^2)^{1/2}$ . The same technique has been also used in the context of safe screening [12]. Although it is possible to consider the intersection of more than two balls, it requires much more tedious algebraic computations.

In a part of the experiments (see § 5), we use a simple but useful trick using the above intersection. When we have a suboptimal solution  $\tilde{w} \in \mathcal{F}$ , we can make use of the center  $m \in \mathcal{F}$  in (6a) as another suboptimal solution, and consider the intersection of the resulting two balls. We show in Lemma 6 in Appendix that the area of the intersection is less than the half of the original ball, meaning that the new bounds could be much tighter than the original ones.

**Kernelization** The bounds (8) in Theorem 2 can be *kernelized*, i.e., what we need to compute in (possibly infinite-dimensional) feature space  $\mathcal{F}$  is only inner products which can be computed by using the associated kernel function  $K$ . The bounds (7) in Theorem 1 can be also kernelized if  $\nabla \ell_i(w), i \in [n]$ , can be kernelized.

## 4 Applications

In this section, we present several practical machine learning tasks in which our algorithmic framework for computing bounds is useful.

### 4.1 Efficient Model Selection

Let us first discuss how our bound computation framework can be used in ordinary model selection scenario. We consider model selection problems for an  $L_2$  regularized convex learning problem in the form of (1). We consider a common situation that two separate training and validation sets are available for training and model selection, respectively. Here, our task is to select the best regularization parameter  $C$  that yields the smallest mis-classification error rate on the validation set among a given list of the candidates  $C_1, \dots, C_T$ . In general, we need to solve all the  $T$  optimization problems for finding the best one<sup>2 3</sup>.

We can use the bounds in Theorem 2 for making the model selection problem more efficient. If we have already computed a solution for a certain  $C_{\tilde{t}}, \tilde{t} \in [T]$ , we can use this solution as the suboptimal solution  $\tilde{w}$ . Then, the lower and the upper bounds of the mis-classification error rate in (2) and (3), respectively, are computed for the remaining candidates. We can make use of the lower bounds for skipping some of the  $T$  training tasks, i.e., if the lower bound of the validation error for a certain  $C_t$  is larger than the smallest validation error obtained so far, we can skip training that model. In addition, these lower and upper bounds are helpful to decide which model should be trained in the next step<sup>4</sup>. A summary of the efficient model selection procedure is described in Algorithm 1.

---

<sup>2</sup> For a certain class of problems, one can compute the exact path of the optimal solutions for the entire range of  $C$ , which is referred to as *regularization path* [9]. Regularization path computation is possible only for a limited class of problems (e.g., it can be computed for an SVM, but not for logistic regression). In addition, regularization path computation is known to be numerically unstable, and does not scale well.

<sup>3</sup> It is also beneficial in practice to use *warm-start* approaches [4] when solving a sequence of optimization problems. For simplicity, we do not take into account the possible advantage of warm-start approach in our discussion here.

<sup>4</sup> In our experiments in § 5, we just selected the  $C_t$  whose validation error lower bound is smallest. There are, however, many other possible approaches. For example, we can select the  $C_t$  whose uncertainty (the difference between the upper and the lower bounds) are largest. See Bayesian optimization for hyperparameter search [14] for detailed discussion on this issue.



---

**Algorithm 1** Efficient Model Selection Algorithm

---

**Input:** training set  $\mathcal{D}_{tr}$ , validation set  $\mathcal{D}_{va}$ , a list of regularization parameters  $\{C_t\}_{t \in [T]}$

**Output:** the optimal solution  $w_{C_{\text{best}}}^*$

```
1:  $\varepsilon_t^\ell \leftarrow 0.0, \varepsilon_t^u \leftarrow 1.0 \ \forall t \in [T]$ 
2:  $\varepsilon_{\text{best}} \leftarrow 1.0, t_{\text{best}} \leftarrow 1, \mathcal{T} \leftarrow [T] \setminus \{1\}$ 
3: while  $\exists t \in \mathcal{T}$  such that  $\varepsilon_t^\ell < \varepsilon_{\text{best}}$  do
4:    $\hat{t} \leftarrow \text{ChooseNextC}(\{\varepsilon_t^\ell, \varepsilon_t^u\}_{t \in \mathcal{T}})$ 
5:    $w_{C_{\hat{t}}}^* \leftarrow \text{TrainModel}(\mathcal{D}_{tr}, C_{\hat{t}})$ 
6:    $\varepsilon_{\hat{t}} \leftarrow \text{ComputeValidError}(w_{C_{\hat{t}}}^*, \mathcal{D}_{va})$ 
7:    $\mathcal{T} \leftarrow \mathcal{T} \setminus \{\hat{t}\}$ 
8:   if  $\varepsilon_{\hat{t}} < \varepsilon_{\text{best}}$  then
9:      $\varepsilon_{\text{best}} \leftarrow \varepsilon_{\hat{t}}, w_{C_{\text{best}}}^* \leftarrow w_{C_{\hat{t}}}^*$ 
10:  end if
11: for  $t \in \mathcal{T}$  do
12:    $\{\tilde{\varepsilon}_t^\ell, \tilde{\varepsilon}_t^u\} \leftarrow \text{ComputeValidErrorBounds}(w_{C_{\hat{t}}}^*, \mathcal{D}_{va}, C_t)$ 
13:    $\varepsilon_t^\ell \leftarrow \max\{\varepsilon_t^\ell, \tilde{\varepsilon}_t^\ell\}, \varepsilon_t^u \leftarrow \min\{\varepsilon_t^u, \tilde{\varepsilon}_t^u\},$ 
14: end for
15: end while
```

---

In Algorithm 1, `ChooseNextC` is a function for selecting one of the remaining regularization parameter  $C_t \in \mathcal{T}$  for the next step. The basic idea here is to select the candidate with which the validation error is expected to be smallest. In this paper, we simply select  $\arg \min_{t \in \mathcal{C}} \varepsilon_t^\ell$  as the next candidate. The function `TrainModel` is used for training the model with the specified regularization parameter. Any specific solvers or general convex optimization tools can be used for this function. The function `ComputeValidError` computes the validation error based on a given solution. The function `ComputeValidErrorBounds` computes the validation error bounds at the specified regularization parameter based on a given solution.

## 4.2 Exact and approximate model selection

**Exact model selection** Although it is common to select the regularization parameter among the finite list of the candidates as we discussed in § 4.1, it would be better if we could find the best possible regularization parameter that exactly minimizes the validation error in the continuous range of  $C \in [C_{\min}, C_{\max}]$ <sup>5</sup>. For the class of  $L_2$ -regularized convex learning problems in the form of (1), such *exact model selections* are possible because we can compute the lower bounds of the validation errors for the continuum of the regularization parameters  $C \in [C_{\min}, C_{\max}]$ .

---

<sup>5</sup> Ideally, we should select the best  $C$  from  $(0, \infty)$ . But it is practically difficult except some special cases. We thus consider selecting  $C$  from an interval between  $C_{\min}$  and  $C_{\max}$ .

Suppose that we have already solved an optimization problem (1) for a certain  $\tilde{C} < C$ , and denote the solution as  $w_{\tilde{C}}^*$ . Then, for an input  $x'_i$  in the validation set, the following rules can be obtained from (8):

$$\begin{aligned}\tilde{C} < C < \frac{\|\phi'_i\| \|w_{\tilde{C}}^*\| + \phi_i'^\top w_{\tilde{C}}^*}{\|\phi'_i\| \|w_{\tilde{C}}^*\| - \phi_i'^\top w_{\tilde{C}}^*} \tilde{C} &\Rightarrow \phi_i'^\top w_{\tilde{C}}^* > 0. \\ \tilde{C} < C < \frac{\|\phi'_i\| \|w_{\tilde{C}}^*\| - \phi_i'^\top w_{\tilde{C}}^*}{\|\phi'_i\| \|w_{\tilde{C}}^*\| + \phi_i'^\top w_{\tilde{C}}^*} \tilde{C} &\Rightarrow \phi_i'^\top w_{\tilde{C}}^* < 0.\end{aligned}$$

Using the above rules, we can compute the lower bounds of the validation errors (2) as a function of  $C \in [C_{\min}, C_{\max}]$ . It means that we can exactly identify a sequence of the regularization parameter values at which the validation error changes by  $1/n'$ . In other words, we can trace all the change points of the validation error along  $C \in [C_{\min}, C_{\max}]$ .

**Model selection with approximation guarantee** The above exact model selection can be relaxed so that it allows to have an  $\varepsilon$ -approximation error, i.e., we can compute a sequence of the models among which there exists a solution whose validation error is within  $\varepsilon$  from the minimum possible value in  $C \in [C_{\min}, C_{\max}]$ . For example, if we set  $\varepsilon$  such that  $\lfloor n'\varepsilon \rfloor = 5$ , then we can compute the sequence of points in  $[C_{\min}, C_{\max}]$  at which the validation error changes by  $5/n'$ .

This model selection scheme can be considered as a valiant of *approximate regularization path* [11, 7, 8]. The key property of these approximate regularization path algorithms is computing the path of solutions with which the approximation error of the objective function values are bounded by  $\varepsilon$ . In our approach, we can control the approximation error of validation performances, which is more useful for model selection purpose.

### 4.3 Fast leave-one-out cross validation

Next, we propose to use our bounds for efficient computation of leave-one-out cross-validation (LOOCV) in binary classification problems. With a slight abuse of notation, let us denote the optimal solution trained with all the instances as  $w_{\text{all}}^*$ , while the optimal solution obtained after picking out an instance  $(x_j, y_j)$  as

$$w_{(-j)}^* := \arg \min_{w \in \mathcal{F}} \frac{1}{2} \|w\|^2 + C \sum_{i \neq j} \ell(y_i, \phi_i^\top w). \quad (9)$$

Then, the LOOCV error is written as

$$\frac{1}{n} \left( \sum_{i: y_i = +1} I(\phi_j^\top w_{(-j)}^* < 0) + \sum_{i: y_i = -1} I(\phi_j^\top w_{(-j)}^* > 0) \right).$$

Our idea here is to compute the bounds of  $\phi_j^\top w_{(-j)}^*$  using  $w_{\text{all}}^*$  as the suboptimal solution for our algorithm. An advantage of this simple approach is that, once we compute  $w_{\text{all}}^*$ , it can be used as the suboptimal solution for bounding all the  $n$  inner products  $\phi_j^\top w_{(-j)}^* \forall j \in [n]$ . If  $\phi_j^\top w_{(-j)}^*$  could be bounded from above or below 0, we do not have to compute the optimal  $w_{(-j)}^*$ . If there are many such instances, the LOOCV computation process would be quite efficient.

## 4.4 Logistic model inference by SVM

Our final application is to make inferences on a model based on a suboptimal model trained by a different learning algorithm. Specifically, we make inferences on a logistic regression model by using the SVM solution trained with the same data set. Logistic regression is especially important and popularly used in biomedical research because the model output and model coefficients are interpreted as the log odds and log odds ratios, respectively. On the other hand, SVM is more popularly used in large-scale machine learning and pattern recognition problems partly because it tends to produce better classification performances and there are many efficient algorithms and solvers that are applicable to large-scale data sets. It is thus important to know how SVM solutions can be useful for inferences on logistic regression models.

Our goal is to make inferences on the solution of the following  $L_2$  regularized logistic regression model

$$w_{\text{LR}}^* := \frac{1}{2} \|w\|^2 + C \sum_{i \in [n]} \log(1 + \exp(-y_i x_i^\top w))$$

by using the suboptimal solution  $\tilde{w} := w_{\text{svm}}^*$  given by

$$w_{\text{svm}}^* := \frac{1}{2} \|w\|^2 + C \sum_{i \in [n]} \max\{0, 1 - y_i x_i^\top w\}.$$

Here, we only consider a linear model, i.e., the feature space  $\mathcal{F}$  is  $d$ -dimensional Euclidean space  $\mathbb{R}^d$ .

Our first interest is in each coefficient of the logistic model solution  $(w_{\text{LR}}^*)_j$  for  $j \in [d]$  because it represents the log odds ratio of the  $j^{\text{th}}$  feature. Using Theorem 1, we can compute the lower and the upper bounds of  $(w_{\text{LR}}^*)_j$  by bounding the inner product  $e(j)^\top w_{\text{LR}}^*$ , where  $e(j)$  is the  $j^{\text{th}}$  coordinate unit vector.

Given the input of a new instance  $x_{\text{new}}$  (e.g., when a new patient profile is provided), our second task is to make an inference on the log odds of the instance. We can compute the lower and the upper bounds of the log odds by bounding the inner product  $x_{\text{new}}^\top w_{\text{LR}}^*$  using Theorem 1.

## 5 Numerical Experiments

In this section, we illustrate the effectiveness of our approach by numerical experiments. We used 12 benchmark data sets listed in Table 1. We used SVM and Logistic Regression (LR) as the two examples of regularized learning problems in the form of (1). LIBSVM [2] and LIBLINEAR [6] were used as the SVM and LR solvers<sup>6</sup>.

We report the results on both linear and nonlinear cases<sup>7</sup>. In nonlinear cases, Gaussian kernel  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$  with  $\gamma = 1/d$  is used.

---

<sup>6</sup> The former provides kernel SVM solver, while the latter provides a linear SVM and a linear LR solvers.

<sup>7</sup> For nonlinear LR, we just used basis expansion approach with Gaussian RBF, and the optimization is conducted by linear LR solver in LIBLINEAR.

Table 1: Datasets used in § 5.

ID	Dataset	$n$	$d$
BCP :	BreastCancerPrognostic	194	33
PKS :	Parkinsons	195	22
SPH :	SPECTHeart	267	44
LVD :	Liver-Disorders	345	6
ION :	Ionosphere	351	33
BCI :	BrainComputerInterface	400	117
BCD :	BreastCancerDiagnostic	569	30
AUS :	Australian	690	14
G2C :	g241c	1,500	241
G2N :	g241n	1,500	241
SPM :	Spambase	4,601	57
MGT :	MAGICGammaTelescope	19,020	10

**Goodness of suboptimal solutions** We conducted simple numerical simulations for understanding the effect of the choice of suboptimal solutions. Figure 1 shows the simulation results of linear LR on two data sets. Here, we randomly generated 1000 suboptimal solutions by adding a Gaussian noise to the optimal solution. The x-axis denotes the distance from the optimal solution  $\|\tilde{w} - w^*\|$ , while the y-axis denotes the tightness of the bounds in (7) measured by the radius  $r$  in (6b). The results indicate that tighter bounds can be obtained as the selected suboptimal solutions approach to the optimal solution.

**Efficient model selection** We examined the efficiency of the model selection strategy discussed in § 4.1. Our task is to find the best regularization parameter  $C$  among  $T = 501$  candidates  $\{C_1, \dots, C_T\}$  evenly allocated between 0.01 and 10000 in logarithmic scale. The basic strategy is to sequentially training the models based on the validation error bounds obtained so far. At each step, we just selected the model that has the smallest validation error lower bound in (2). In this experimental setup, we have multiple trained models that can be used as the suboptimal models. We thus used the closest two models (one with smaller  $C$  and the other with larger  $C$ ) as the suboptimal models, and employed the intersection approach discussed in § 3. Figure 2 shows the validation error bounds after the last step where we could find the best regularization parameter. Table 2 shows how many training optimization problems were solved before finding the best one. The results indicate that the best regularization parameter can be found without solving all the  $T = 501$  optimization problems.

**Exact and approximate model selection** We examined the effectiveness of the *exact* and *approximate* model selection schemes discussed in § 4.2. We set  $C_{\min} = 0.01$  and  $C_{\max} = 100$ . The task of *exact model selection* is to find the best possible regularization parameter that exactly minimizes the validation error in

Table 2: The number of optimization problems solved before finding the best regularization parameter (among the 501 models).

Linear Model			Nonlinear Model		
Data	LR	SVM	Data	LR	SVM
BCP	421/501	196/501	BCP	321/501	56/501
LVD	274/501	122/501	PKS	366/501	58/501
ION	98/501	151/501	SPH	381/501	74/501
G2C	337/501	99/501	BCD	336/501	54/501

the continuous range of  $C \in [C_{\min}, C_{\max}]$ . On the other hand, in  $\varepsilon$ -approximate model selection scheme, we can find a solution whose validation error is shown to be within  $\varepsilon$  from the minimum possible value in that range. Starting from  $C = C_{\min}$ , we gradually increased the regularization parameter  $C$  so that the change of the validation errors are within  $\varepsilon \in \{0, 0.01, 0.05, 0.1\}$ . The results shown in Figure 3 and Table 3 indicate that the number of models we need to train decreases as  $\varepsilon$  increases.

Table 3: Experimental results on the exact and approximate model selection schemes. The numbers in the table represent how many models were solved in the path.

Linear LR	$\varepsilon = 0.1$	0.05	0.01	0 (exact)
ION	86	205	1646	13839
BCD	33	66	211	2654
Linear SVM	0.1	0.05	0.01	0 (exact)
ION	107	230	2390	17592
BCD	37	77	468	8817
Nonlinear LR	0.1	0.05	0.01	0 (exact)
BCP	341	633	19956	19956
PKS	292	523	18939	18939
Nonlinear SVM	0.1	0.05	0.01	0 (exact)
BCP	293	711	9365	9365
PKS	204	428	19768	19768

**Fast LOOCV** We investigated the efficiency of LOOCV computation in linear LR. We compared a naive approach (full) and the proposed approach (proposed). In the naive approach,  $n$  optimization problems in the form of (9) were solved after removing each of the  $n$  instances. In the proposed approach, we first computed the model  $w_{\text{all}}^*$  by solving the training optimization problem with all the  $n$  instances. Then, the lower and the upper bounds of  $\phi(x_j)^\top w_{(-j)}^*$  were computed based on Theorem 1 by using  $w_{\text{all}}^*$  as our choice of the suboptimal model. If the lower bound was larger than 0 or the upper bound was smaller than 0,

we skipped solving the optimization problem (9) for that instance. Figure 4 and Table 4 show the results. Figure 4 indicates that we could skip solving the optimization problem for many instances especially when the regularization parameter  $C$  is small. From the results in Table 4, we could see that the costs of computing the lower and the upper bounds are negligible compared with the cost of solving optimization problems.

**LR inference by SVM** Finally, we present a numerical illustration of LR inferences based on an SVM solution as discussed in § 4.4. In Figure 5, the blue circles and the green diamonds represent the SVM coefficients  $w_{\text{SVM}}^*$  and the optimal LR coefficients  $w_{\text{LR}}^*$ , respectively. The blue bars indicate the lower and the upper bounds of the optimal LR coefficients obtained by using the optimal SVM solution as our choice of the suboptimal model. The top plot is the result obtained by applying Theorem 1, while the bottom plot is the result after considering the intersection of the two balls as described in § 3. The results indicate the advantage of using such an intersection.

## 6 Conclusions

In this paper, we introduced a novel algorithmic framework for computing the lower and the upper bounds of the quantities depending on the unknown optimal solution. Although we mainly focused on model selection for binary classification problems in this paper, our framework can be used in many other machine learning problems. For example, we can easily extend our results to LASSO problem (see Appendix B for details). As we discussed, the choice of the suboptimal model  $\tilde{w}$  is critically important for obtaining useful tight bounds. An important future work is to develop an algorithm for finding a good suboptimal model.

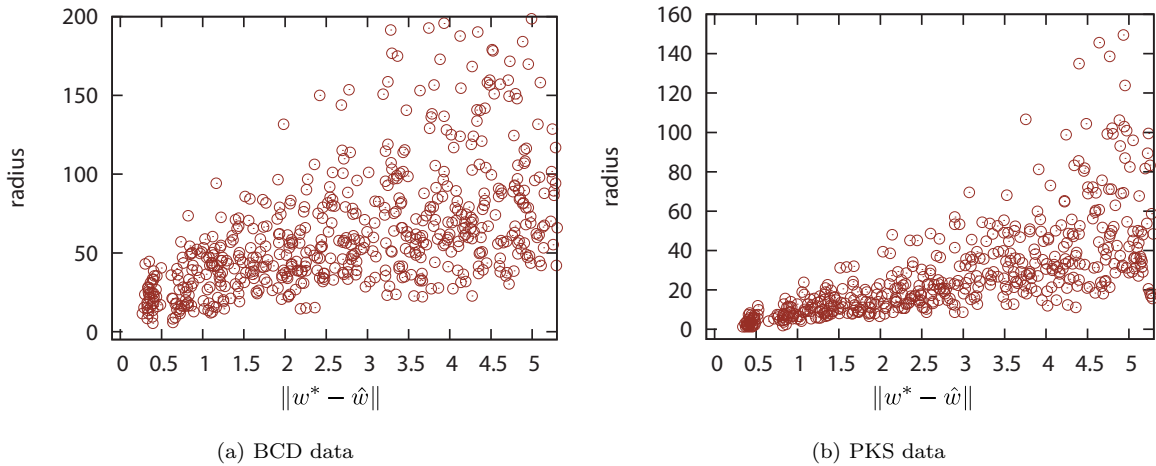


Figure 1: The simulation results for understanding the effects of the suboptimal model on two data sets.

Table 4: The computational time [sec] of LOOCV computation in the naive approach (full) and the proposed approach (proposed) for  $C \in \{0.01, 1, 100\}$ . The numbers in the parenthesis are the time taken for computing the lower and the upper bounds in Theorem 1.

PKS	$C = 0.01$	$C = 1$	$C = 100$
full	2.15	1.17	1.79
proposed(bounds)	<b>0.41</b> (0.22)	<b>0.37</b> (0.01)	<b>1.44</b> (0.02)
relative costs	0.19	0.31	0.80
BCI	$C = 0.01$	$C = 1$	$C = 100$
full	6.64	13.98	39.43
proposed(bounds)	<b>2.58</b> (0.13)	<b>9.39</b> (0.04)	<b>24.39</b> (0.06)
relative costs	0.38	0.67	0.61
BCD	$C = 0.01$	$C = 1$	$C = 100$
full	3.94	3.57	7.49
proposed(bounds)	<b>0.42</b> (0.19)	<b>0.19</b> (0.02)	<b>1.04</b> (0.03)
relative costs	0.10	0.053	0.13
AUS	$C = 0.01$	$C = 1$	$C = 100$
full	3.5	3.48	4.01
proposed(bounds)	<b>0.24</b> (0.13)	<b>0.6</b> (0.05)	<b>3.32</b> (0.03)
relative costs	0.068	0.17	0.82
G2C	$C = 0.01$	$C = 1$	$C = 100$
full	84.04	192.94	292.26
proposed(bounds)	<b>11.98</b> (0.46)	<b>62.5</b> (0.40)	<b>127</b> (0.35)
relative costs	0.14	0.32	0.43
G2N	$C = 0.01$	$C = 1$	$C = 100$
full	104.73	220.34	358.82
proposed(bounds)	<b>13.87</b> (0.37)	<b>70.7</b> (0.34)	<b>141.71</b> (0.39)
relative costs	0.13	0.32	0.39
SPM	$C = 0.01$	$C = 1$	$C = 100$
full	268.68	794.02	2783.59
proposed(bounds)	<b>3.71</b> (0.98)	<b>180.00</b> (1.14)	<b>1761.4</b> (1.11)
relative costs	0.013	0.22	0.63
MGT	$C = 0.01$	$C = 1$	$C = 100$
full	1043.7	1033.45	1064.65
proposed(bounds)	<b>20.43</b> (7.16)	<b>321.17</b> (7.34)	<b>782.84</b> (6.19)
relative costs	0.019	0.31	0.73

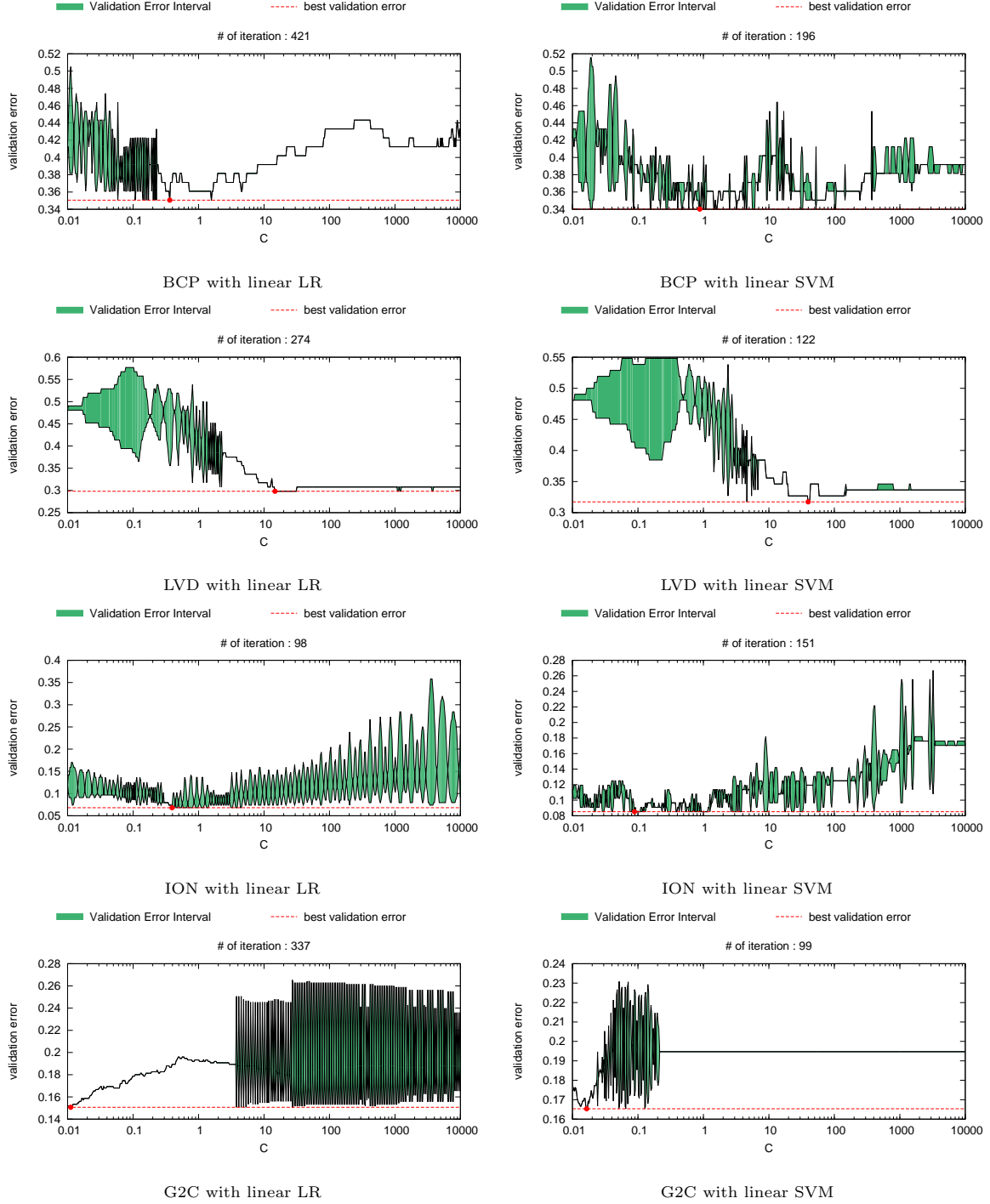
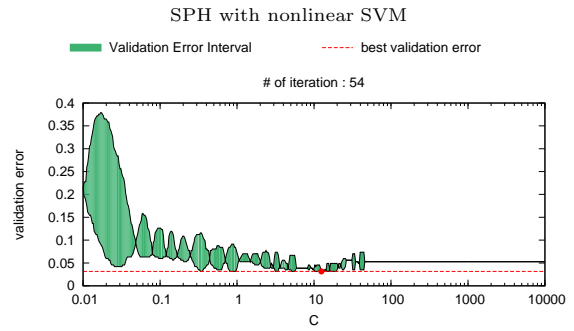
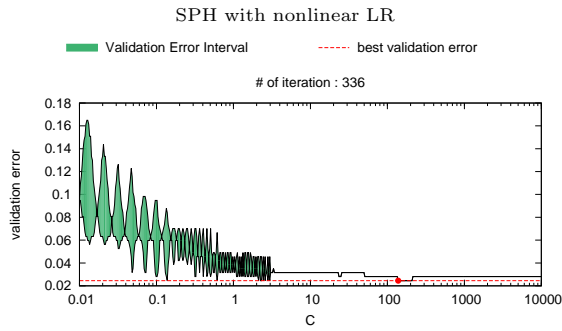
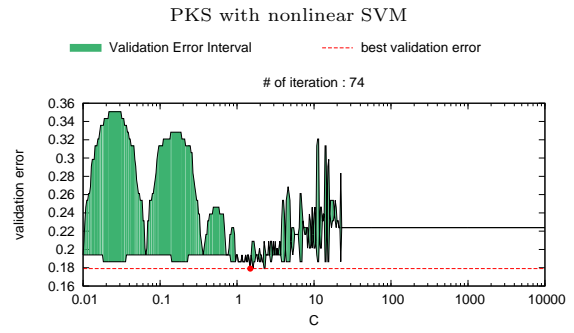
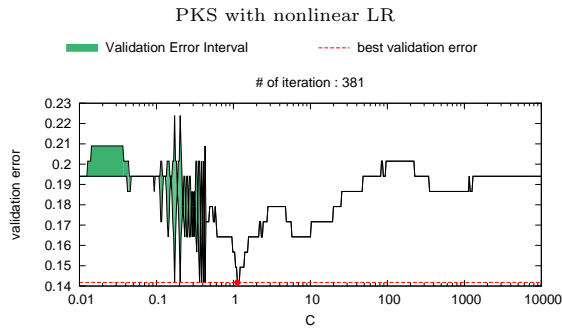
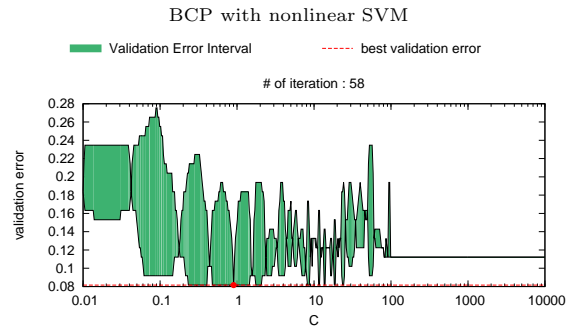
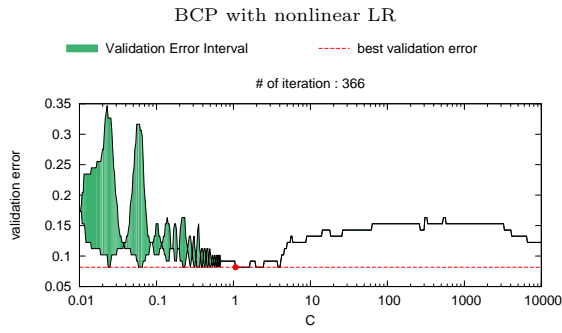
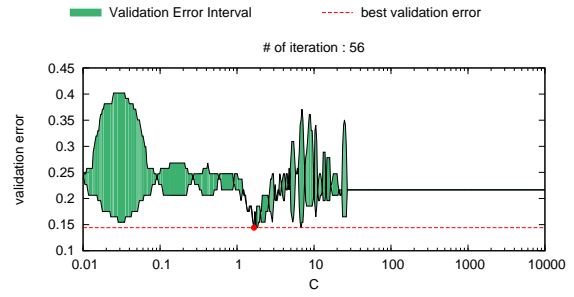
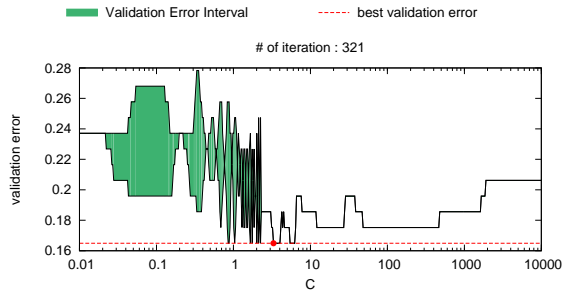


Figure 2: The sequence of validation error bounds after the final step of the efficient model selection processes. Although the validation errors with several regularization parameters are still unknown, we can guarantee that the current smallest solution (red point) is the best one.





BCD with nonlinear LR

BCD with nonlinear SVM

Figure 2: Continued.

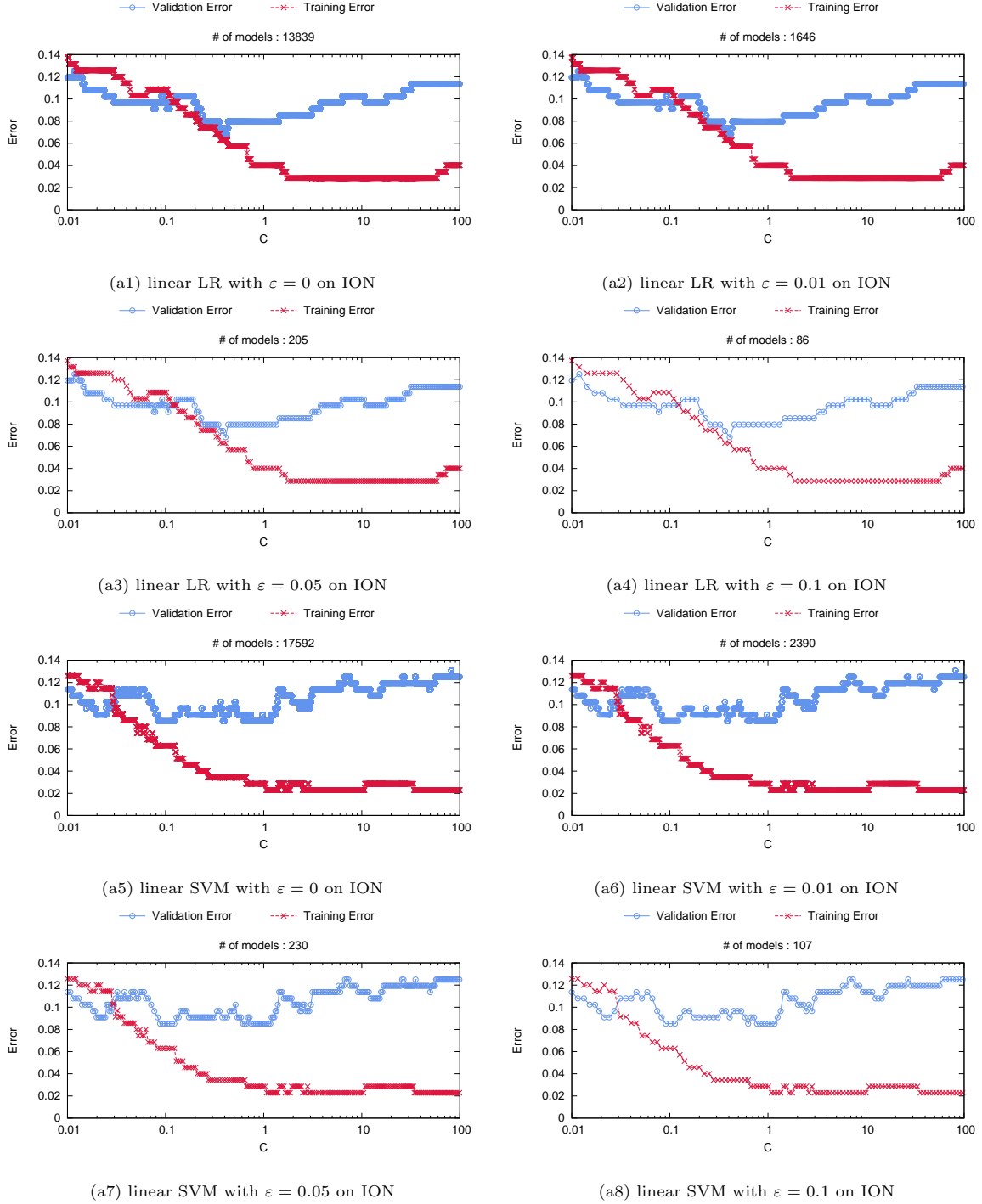
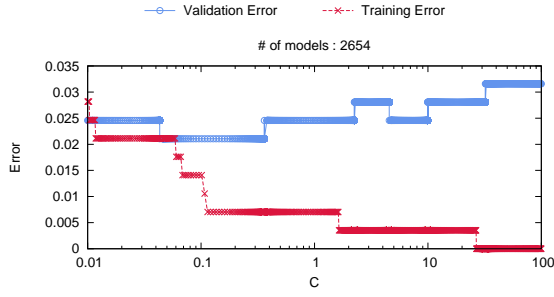
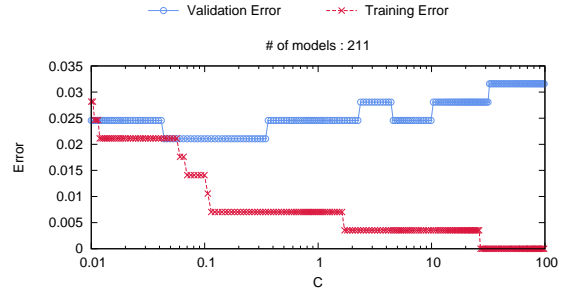


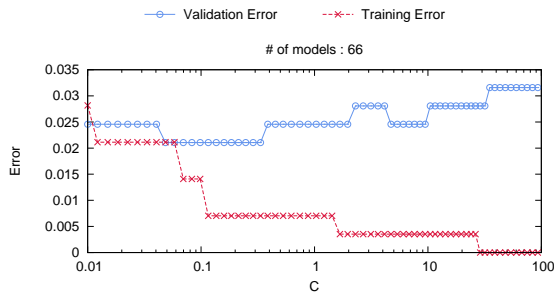
Figure 3: Exact and approximate model selection results. The validation error of the solution is shown to be within  $\varepsilon$  from the smallest possible value in the continuous range of  $C \in [C_{\min}, C_{\max}]$ .



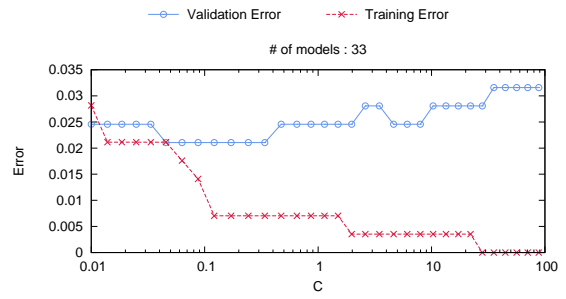
(b1) linear LR with  $\varepsilon = 0$  on BCD



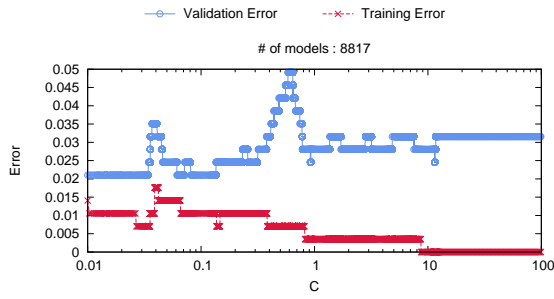
(b2) linear LR with  $\varepsilon = 0.01$  on BCD



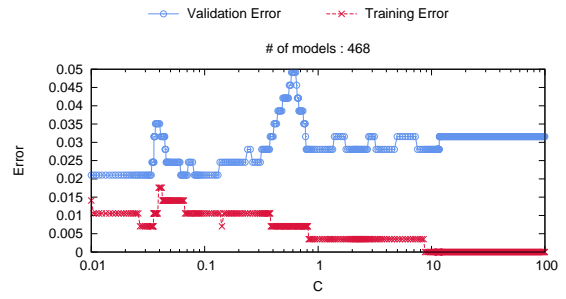
(b3) linear LR with  $\varepsilon = 0.05$  on BCD



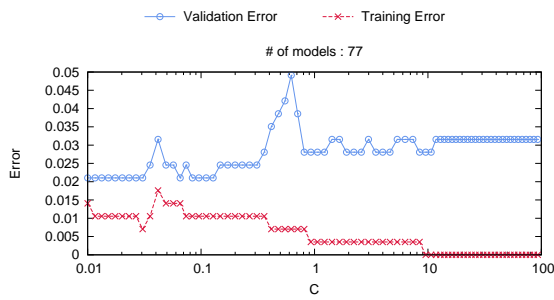
(b4) linear LR with  $\varepsilon = 0.1$  on BCD



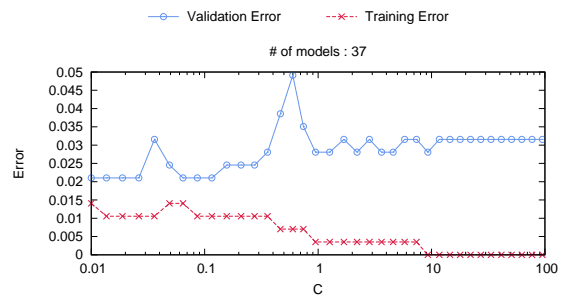
(b5) linear SVM with  $\varepsilon = 0$  on BCD



(b6) linear SVM with  $\varepsilon = 0.01$  on BCD

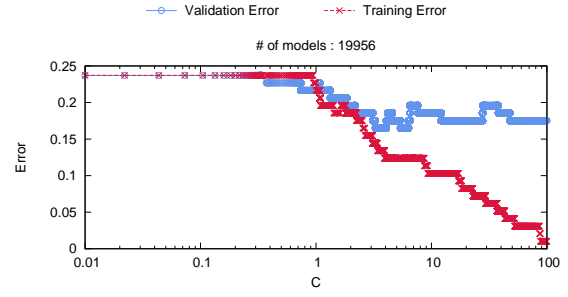
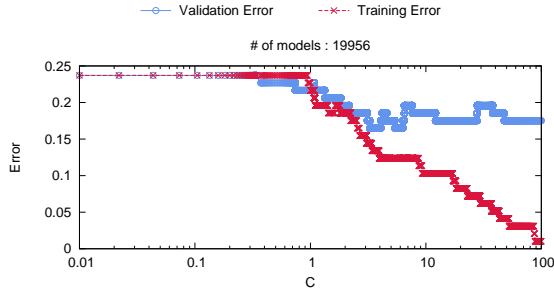


(b7) linear SVM with  $\varepsilon = 0.05$  on BCD



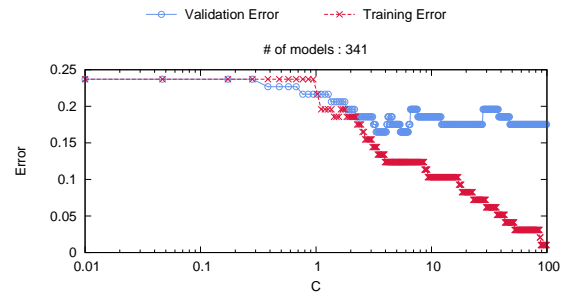
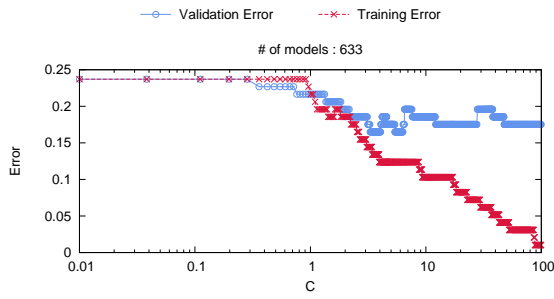
(b8) linear SVM with  $\varepsilon = 0.1$  on BCD

Figure 3: Continued.



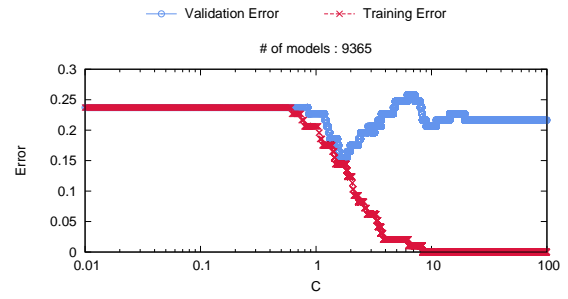
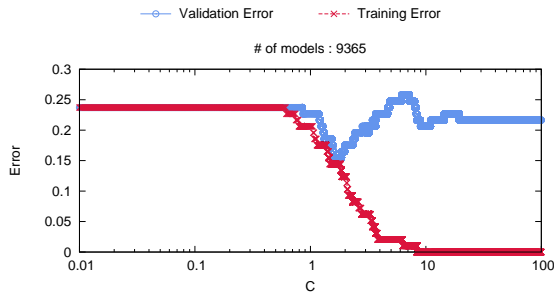
(c1) nonlinear LR with  $\varepsilon = 0$  on BCP

(c2) nonlinear LR with  $\varepsilon = 0.01$  on BCP



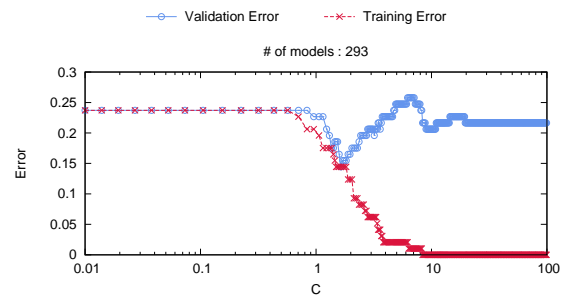
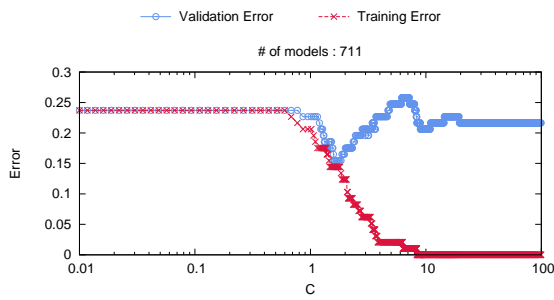
(c3) nonlinear LR with  $\varepsilon = 0.05$  on BCP

(c4) nonlinear LR with  $\varepsilon = 0.1$  on BCP



(c5) nonlinear SVM with  $\varepsilon = 0$  on BCP

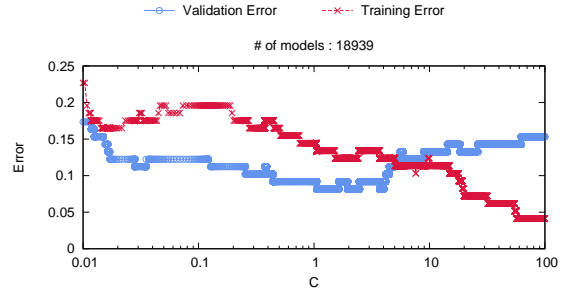
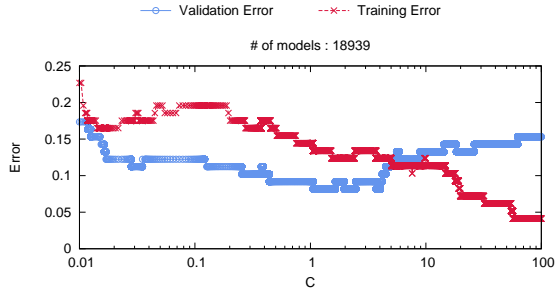
(c6) nonlinear SVM with  $\varepsilon = 0.01$  on BCP



(c7) nonlinear SVM with  $\varepsilon = 0.05$  on BCP

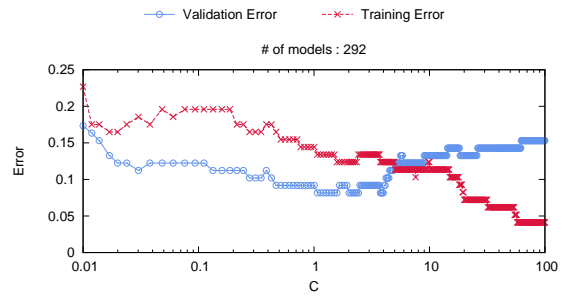
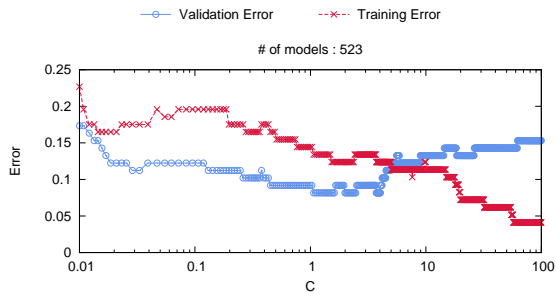
(c8) nonlinear SVM with  $\varepsilon = 0.1$  on BCP

Figure 3: Continued.



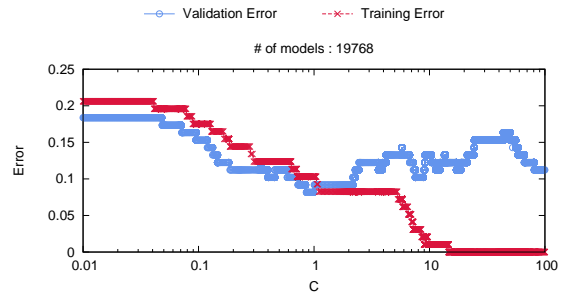
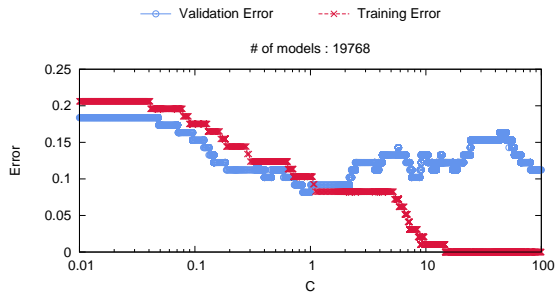
(d1) nonlinear LR with  $\varepsilon = 0$  on BCP

(d2) nonlinear LR with  $\varepsilon = 0.01$  on BCP



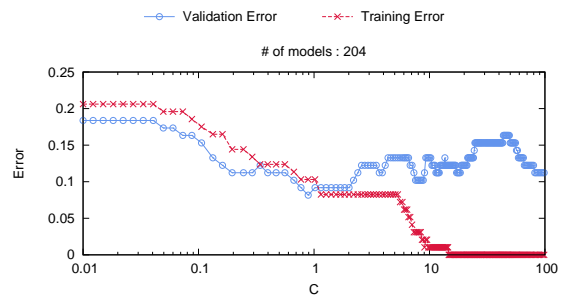
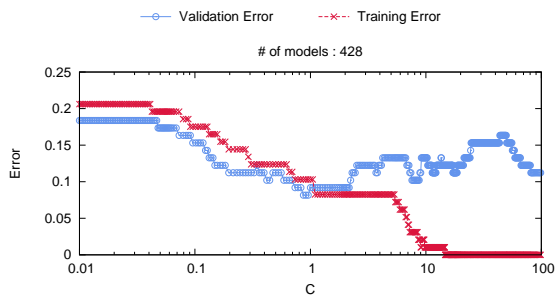
(d3) nonlinear LR with  $\varepsilon = 0.05$  on BCP

(d4) nonlinear LR with  $\varepsilon = 0.1$  on BCP



(d5) nonlinear SVM with  $\varepsilon = 0$  on BCP

(d6) nonlinear SVM with  $\varepsilon = 0.01$  on BCP



(d7) nonlinear SVM with  $\varepsilon = 0.05$  on BCP

(d8) nonlinear SVM with  $\varepsilon = 0.1$  on BCP

Figure 3: Continued.

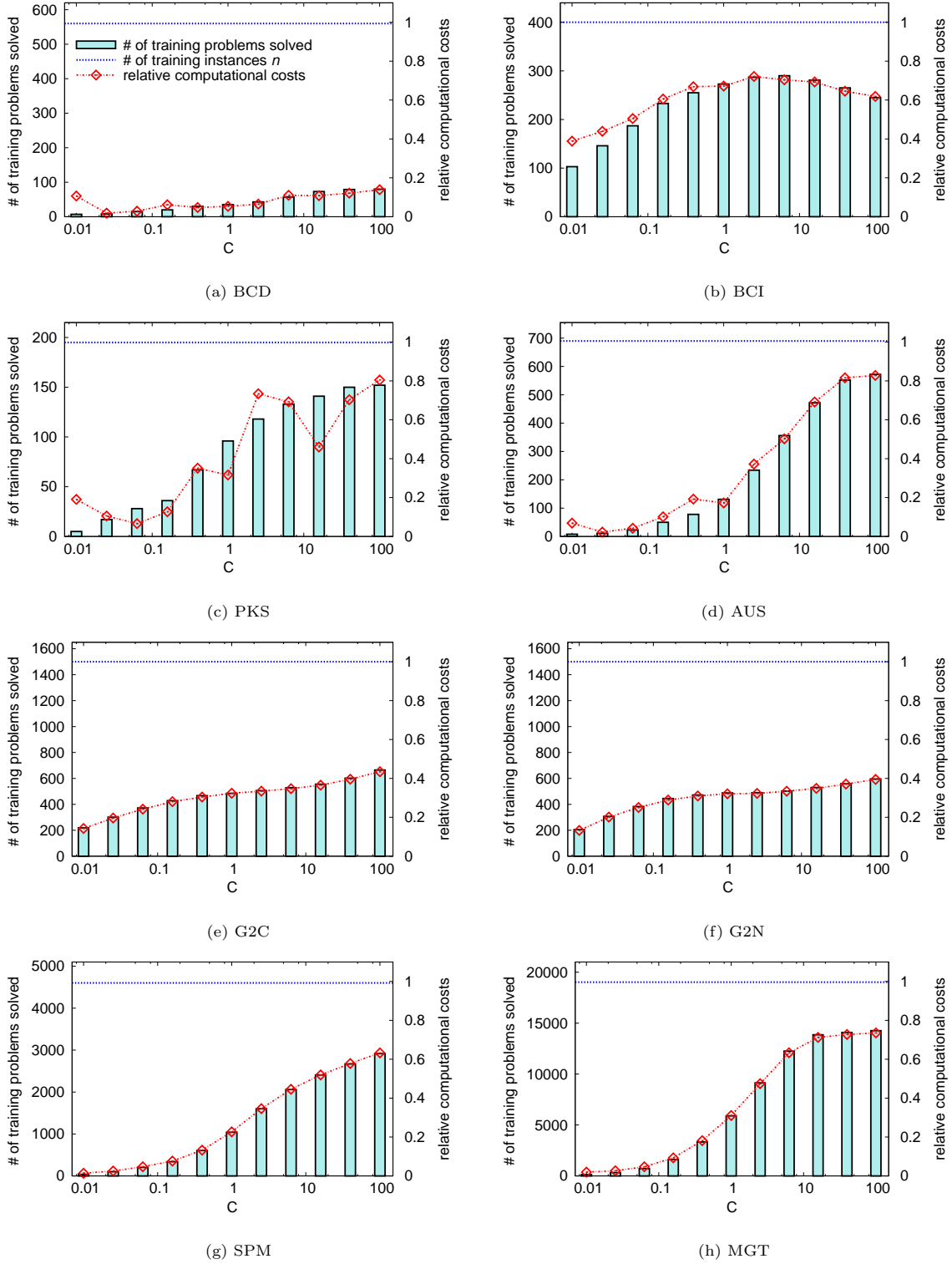
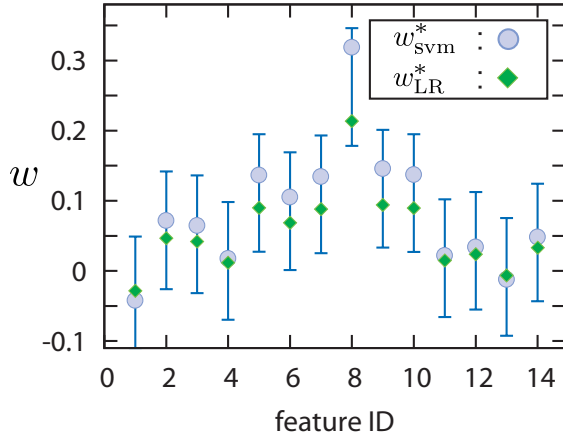
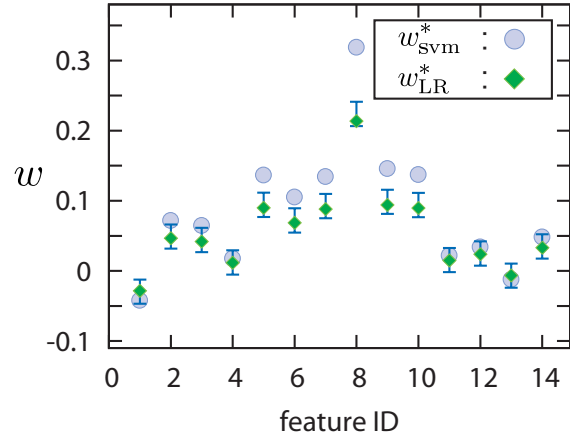


Figure 4: The results on fast LOOCV computation experiments. The number of optimization problems solved in our proposed approach (light blue bars) and the relative computational costs (red dotted lines) are shown.



(a) A single ball



(b) Intersection of two balls

Figure 5: The lower and the upper bounds of coefficients  $w_{\text{LR}}^*$  obtained by using the SVM solution as the suboptimal model. (a) The bounds were computed based on a single ball in Theorem 1. (b) The bounds were computed based on the intersection of the two balls as described in § 3.

## References

- [1] D. P. Bertsekas. *Nonlinear Programming (2nd edition)*. Athena Scientific, 1999.
- [2] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
- [3] L. Dai and K. Pelckmans. An ellipsoid based two-stage sreening test for bpdn. In *Proceedings of the 20th European Signal Processing Conference*, 2012.
- [4] D. DeCoste and K. Wagstaff. Alpha seeding for support vector machines. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.
- [5] L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination in sparse supervised learning. *Pacic Journal of Optimization*, 8:667–698, 2012.
- [6] R. R. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [7] J. Giesen, M. Jaggi, and S. Laue. Approximating parameterized convex optimization problems. *ACM Transactions on Algorithms*, 9, 2012.
- [8] J. Giesen, J. Mueller, S. Laue, and S. Swiercy. Approximating concavely parameterized optimization problems. In *Advances in Neural Information Processing Systems 25*, pages 2114–2122, 2012.
- [9] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–415, 2004.
- [10] J. Liu, Z. Zhao, J. Wang, and J. Ye. Safe screening with variational inequalities and its application to lasso. *arXiv:1307.7577*, 2013.
- [11] J. Mairal and B. Yu. Complexity analysis of the LASSO regularization path. In *Proceedings of the 29th International Conference on Machine Learning*, pages 79–186, 2012.
- [12] K. Ogawa, Y. Suzuki, S. Suzumura, and I. Takeuchi. Safe sample screening for support vector machines. *arXiv:1401.6740*, 2014.
- [13] K. Ogawa, Y. Suzuki, and I. Takeuchi. Safe screening of non-support vectors in pathwise SVM computation. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.
- [14] J. Sneek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 2012*, 2012.
- [15] J. Wang, B. Lin, P. Gong, P. Wonka, and J. Ye. Lasso screening rules via dual polytope projection. *arXiv:1211.3966*, 2012.



- [16] J. Wang, J. Liu, and J. Ye. Efficient mixed-norm regularization: Algorithms and safe screening methods. *arXiv:1307.4156*, 2013.
- [17] J. Wang, J. Zhou, J. Liu, P. Wonka, and J. Ye. A safe screening rule for sparse logistic regression. *arXiv:1307.4152*, 2013.
- [18] Y. Wang, Z. J. Xiang, and P. J. Ramadge. Lasso screening with a small regularization parameters. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013.
- [19] Y. Wang, Z. J. Xiang, and P. J. Ramadge. Tradeoffs in improved screening of lasso problems. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013.
- [20] H. Wu and P. J. Ramadge. The 2-codeword screening test for lasso problems. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013.
- [21] Z. J. Xiang and P. J. Ramadge. Fast lasso screening test based on correlatins. In *Proceedings of the 37th IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012.
- [22] Z. J. Xiang, H. Xu, and P. J. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *Advances in Neural Information Processing Systems 24*, pages 900–908, 2012.

## A Proofs

Let us first clarify the optimality condition of a convex constrained optimization problem in the following proposition.

**Proposition 4.** *Consider the following general problem:*

$$\min_z g(z) \quad \text{s.t. } z \in \mathcal{Z}, \quad (10)$$

where  $g$  is a convex differentiable function and  $\mathcal{Z}$  is a convex set. Then a solution  $z^*$  is the optimal solution of (10) if and only if

$$\nabla g(z^*)^\top (z^* - z) \leq 0 \quad \forall z \in \mathcal{Z},$$

where  $\nabla g(z^*)$  is the gradient vector of  $g$  at  $z = z^*$ .

See, for example, Proposition 2.1.2 in [1] for the proof of Proposition 4.

*Proof of Theorem 1.* Let us first rewrite the problem (1) by using a slack variable  $\xi \in \mathbb{R}$  as

$$\min_{w \in \mathcal{F}, \xi \in \mathbb{R}} J_C(w, \xi) := \frac{1}{2} \|w\|^2 + C\xi \quad (11a)$$

$$\text{s.t. } \xi \geq \sum_{i \in [n]} \ell_i(w). \quad (11b)$$

It is easy to see that the optimal solution of the problem (11) is  $w = w_C^*$  and  $\xi = \xi_C^* := \sum_{i \in [n]} \ell_i(w_C^*)$ .

Noting that  $(\tilde{w}, \tilde{\xi})$  with  $\tilde{\xi} := \sum_{i \in [n]} \ell_i(\tilde{w})$  is a feasible solution of (11), from Proposition 4,

$$\nabla J_C(w_C^*, \xi_C^*)^\top \left( \begin{bmatrix} w_C^* \\ \xi_C^* \end{bmatrix} - \begin{bmatrix} \tilde{w} \\ \tilde{\xi} \end{bmatrix} \right) \leq 0, \quad (12)$$

where  $\nabla J_C(w_C^*, \xi_C^*)$  is the gradient vector of  $J_C$  at  $(w, \xi) = (w_C^*, \xi_C^*)$ . By substituting  $\nabla J_C(w_C^*, \xi_C^*) = (w_C^{*\top}, C)^\top$  into (12), it is written as the following quadratic inequality constraint:

$$\|w_C^*\|^2 - \tilde{w}^\top w_C^* + C(\xi_C^* - \sum_{i \in [n]} \ell_i(\tilde{w})) \leq 0. \quad (13)$$

On the other hand, the constraint (11b) indicates that the optimal solution  $(w_C^*, \xi_C^*)$  satisfies the following linear inequality constraint:

$$\xi_C^* \geq \sum_{i \in [n]} \ell_i(w_C^*) \geq \sum_{i \in [n]} (\ell_i(\tilde{w}) + \nabla \ell_i(\tilde{w})^\top (w_C^* - \tilde{w})), \quad (14)$$

where  $\nabla \ell_i(\tilde{w})$  is the gradient vector (or a subgradient vector in non-differentiable case) of  $\ell_i$  at  $w$ . Here, note that, the second inequality follows from the assumption that  $\ell_i$  is convex, and the last line is the tangent hyperplane (or a supporting hyperplane in non-differentiable case) of  $\ell_i$  at  $\tilde{w}$ . By combining (13) and (14), we have

$$\|w_C^* - \frac{1}{2} \left( \tilde{w} - C \sum_{i \in [n]} \nabla \ell_i(\tilde{w}) \right)\|^2 \leq \left\{ \frac{1}{2} \left\| \tilde{w} + C \sum_{i \in [n]} \nabla \ell_i(\tilde{w}) \right\| \right\}^2 \Leftrightarrow \|w_C^* - m\|^2 \leq r^2, \quad (15)$$

where  $m \in \mathcal{F}$  and  $r \geq 0$  are defined in (6).

Since (15) indicates that the optimal solution  $w_C^*$  is within the ball

$$\mathcal{S} := \{w \mid \|w - m\| \leq r\}, \quad (16)$$

the problem of computing the lower bound of  $\theta^\top w_C^*$  is formulated as

$$b_{lo}(\theta^\top w_C^*) = \min_{w \in \mathcal{S}} \theta^\top w. \quad (17)$$

Using the standard Lagrange multiplier theory, the solution of the problem

$$\min_w \theta^\top w \quad \text{s.t.} \quad \|w - m\|^2 \leq r^2$$

can be explicitly solved as

$$b_{lo}(\theta^\top w_C^*) = \min_{w \in \mathcal{S}} \theta^\top w = \theta^\top m - \|\theta\|r.$$

The upper bound of  $\theta^\top w_C^*$  is similarly obtained as

$$b_{up}(\theta^\top w_C^*) = \max_{w \in \mathcal{S}} \theta^\top w = \theta^\top m + \|\theta\|r.$$

□

*Proof of Theorem 2.* We first consider a case where the loss functions  $\ell_i, i \in [n]$ , are differentiable at  $w = w_C^*$ . In this case, we can easily prove the theorem just by substituting  $w_C^*$  into  $\tilde{w}$  and use the proof of Theorem 1. Specifically, since  $w_C^*$  minimizes  $\frac{1}{2}\|w\|^2 + \tilde{C} \sum_{i \in [n]} \ell_i(w)$ , the gradient at  $w = w_C^*$  is zero, i.e.,

$$\left. \frac{\partial}{\partial w} \left( \frac{1}{2}\|w\|^2 + \tilde{C} \sum_{i \in [n]} \ell_i(w) \right) \right|_{w=w_C^*} = 0 \Leftrightarrow w_C^* + \tilde{C} \sum_{i \in [n]} \nabla \ell_i(w_C^*) = 0 \Leftrightarrow \sum_{i \in [n]} \nabla \ell_i(w_C^*) = -\frac{1}{\tilde{C}} w_C^*. \quad (18)$$

Thus, in this case, the center  $m \in \mathcal{F}$  and the radius  $r > 0$  in (6) are written as

$$m = \frac{C + \tilde{C}}{2\tilde{C}} w_C^* \text{ and } r = \frac{|C - \tilde{C}|}{2\tilde{C}} \|w_C^*\|. \quad (19)$$

By substituting (19) into (7), we have the bounds in the form of (8).

Next, we consider a case where the loss function is not differentiable at  $w = w_C^*$ . Noting that  $w_C^*$  is the optimal solution, from Proposition 4,

$$\nabla J_{\tilde{C}}(w_C^*, \xi_C^*)^\top \left( \begin{bmatrix} w_C^* \\ \xi_C^* \end{bmatrix} - \begin{bmatrix} \hat{w} \\ \hat{\xi} \end{bmatrix} \right) \leq 0 \quad \text{for any } \hat{w} \in \mathcal{F}, \quad (20)$$

where we defined  $\xi_C^* := \sum_{i \in [n]} \ell_i(w_C^*)$  and  $\hat{\xi} := \sum_{i \in [n]} \ell_i(\hat{w})$ . Since it can be rewritten as

$$\sum_{i \in [n]} \ell_i(\hat{w}) \geq \sum_{i \in [n]} \ell_i(w_C^*) + \left( -\frac{1}{\tilde{C}} w_C^* \right)^\top (\hat{w} - w_C^*) \quad \text{for any } \hat{w} \in \mathcal{F}, \quad (21)$$

we see that  $-\frac{1}{\tilde{C}}w_{\tilde{C}}^* \in \mathcal{F}$  is a subgradient vector of  $\sum_{i \in [n]} \ell_i(w)$  at  $w = w_{\tilde{C}}^*$ , i.e., we can replace  $\sum_{i \in [n]} \nabla \ell_i(\tilde{w})$  in (14) with  $-\frac{1}{\tilde{C}}w_{\tilde{C}}^*$  when  $\tilde{w} = w_{\tilde{C}}^*$ . If we set  $\hat{w} := w_C^*$  in (21), we have the following linear inequality constraint on  $w_C^*$ :

$$\xi_C^* \geq -\frac{1}{\tilde{C}}w_{\tilde{C}}^{*\top}w_C^* + \frac{1}{\tilde{C}}\|w_{\tilde{C}}^*\|^2 + \sum_{i \in [n]} \ell_i(w_{\tilde{C}}^*). \quad (22)$$

By combining (13) with  $\tilde{w} := w_{\tilde{C}}^*$  and (22), we have

$$\|w_C^* - \frac{C + \tilde{C}}{2\tilde{C}}w_{\tilde{C}}^*\|^2 \leq \left\{ \frac{|C - \tilde{C}|}{2\tilde{C}}\|w_{\tilde{C}}^*\| \right\}^2. \quad (23)$$

It indicates that the optimal solution  $w_C^*$  is in the ball with the center and the radius defined by (19).  $\square$

## B Additional Theoretical Results

**Bounding Lasso dual solutions** We can easily confirm that the lower and the upper bounds (7) in Theorem 1 are still true when we have some additional convex constraints in (1). The following theorem tells that we can obtain similar bounds for LASSO problem.

**Theorem 5.** *Let us consider a regression problem with the training set  $\{(x_i, y_i)\}_{i \in [n]}$ ,  $x_i \in \mathbb{R}^d$ ,  $y_i \in \mathbb{R}$ . We denote the  $n \times d$  input (design) matrix as  $X := [z_1 \ \dots \ z_d] \in \mathbb{R}^{n \times d}$ , where  $z_j$  represents the  $j^{\text{th}}$  column of  $X$ , and the  $n$ -dimensional output (target) vector as  $y := [y_1, \dots, y_n]^\top \in \mathbb{R}^n$ .*

*A well-known Lasso problem is formulated as*

$$\beta_\lambda^* := \arg \min_{\beta \in \mathbb{R}^d} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|, \quad (24)$$

*where  $\lambda > 0$  is the regularization parameter. The dual of (24) is written as*

$$\alpha_\lambda^* := \arg \min_{\alpha \in \mathbb{R}^n} \frac{1}{2}\|\alpha - \frac{1}{\lambda}y\|^2 \quad \text{s.t.} \quad \|X^\top \alpha\|_\infty \leq 1, \quad (25)$$

*where  $\alpha \in \mathbb{R}^n$  is the Lagrange multipliers. Then, for any  $\theta \in \mathbb{R}^n$ , the inner product  $\theta^\top \alpha_\lambda^*$  is bounded as*

$$\theta^\top m_{\text{Lasso}} - \|\theta\|r_{\text{Lasso}} \leq \theta^\top \alpha_\lambda^* \leq \theta^\top m_{\text{Lasso}} + \|\theta\|r_{\text{Lasso}},$$

*where  $m_{\text{Lasso}} \in \mathbb{R}^d$  and  $r_{\text{Lasso}} > 0$  are defined, with any  $\tilde{\alpha} \in \mathbb{R}^n$ , as*

$$m_{\text{Lasso}} := \frac{1}{2}(\tilde{\alpha} + \frac{1}{\lambda}y), \quad r_{\text{Lasso}} := \frac{1}{2}\|\tilde{\alpha} - \frac{1}{\lambda}y\|. \quad (26)$$

*Proof.* The dual problem (25) is rewritten as

$$\alpha_\lambda^* := \arg \min_{\alpha \in \mathbb{R}^n} \frac{1}{2}\|\alpha - \frac{1}{\lambda}y\|^2 \quad \text{s.t.} \quad \|X^\top \alpha\|_\infty \leq 1 = \arg \min_{\alpha \in \mathbb{R}^n} \frac{1}{2}\|\alpha\|^2 - \frac{1}{\lambda} \sum_{i \in [n]} \alpha_i y_i \quad \text{s.t.} \quad \|X^\top \alpha\|_\infty \leq 1. \quad (27)$$

Noting that (27) has the same form as our  $L_2$  regularized learning problem in (1), we can similarly compute the lower and the upper bounds of the inner product  $\theta^\top \alpha_\lambda^*$ .  $\square$

An important consequence of Theorem 5 is that, the lower and the upper bounds of the (negative) residual of the Lasso  $x_i^\top \beta_\lambda^* - y_i, i \in [n]$ , can be obtained by using the relationship:

$$(\alpha_\lambda^*)_i \equiv f(x_i) - y_i, \quad i \in [n]. \quad (28)$$

Specifically, the residual is bounded as

$$-(m_{\text{Lasso}})_i - r_{\text{Lasso}} \leq y_i - x_i^\top \beta_\lambda^* \leq -(m_{\text{Lasso}})_i + r_{\text{Lasso}}. \quad (29)$$

Another important relationship in Lasso is

$$|z_j^\top \alpha_\lambda^*| < 1 \Rightarrow (\beta_\lambda^*)_j = 0, \quad j \in [d]. \quad (30)$$

Using our bounds, it indicates that

$$\max \left\{ \left| z_j^\top m_{\text{Lasso}} - \|z_j\| r_{\text{Lasso}} \right|, \left| z_j^\top m_{\text{Lasso}} + \|z_j\| r_{\text{Lasso}} \right| \right\} < 1 \Rightarrow (\beta_\lambda^*)_j = 0, \quad (31)$$

i.e., the  $j^{\text{th}}$  variable can be removed without actually computing the optimal solution  $\beta_\lambda^*$ . This computational trick is called *safe screening* and has been intensively studied in the literature [5, 22, 21, 3, 15, 13, 18, 19, 20, 16, 17, 12]. Actually, we can easily show that our ball defined in Theorem 5 is equivalent to (14) in [10]. In this sense, our results in Theorems 1 and 5 are considered as the general form that includes safe screening as a special case.

**How to find small intersection of two balls** In § 3, we slightly mentioned about a simple trick for finding a small intersection of two balls in which the optimal solution  $w_C^*$  is guaranteed to exist. When we have a suboptimal model  $\tilde{w} \in \mathcal{F}$ , our idea is to make use of the center  $m \in \mathcal{F}$  in (6a) as another suboptimal solution, and consider the intersection of the resulting two balls. The following lemma indicates that the volume of the intersection is at most half of the original ball.

**Lemma 6.** *For any  $\tilde{w} \in \mathcal{F}$ , let  $\{\tilde{w}_t \in \mathcal{F}\}_{t \in \mathbb{N}}$  be the series of vectors defined by*

$$\tilde{w}_1 := \tilde{w} \text{ and } \tilde{w}_{t+1} = \frac{1}{2} \left( \tilde{w}_t - C \sum_{i \in [n]} \nabla \ell_i(\tilde{w}_t) \right) \quad \forall t \geq 1.$$

*Furthermore, let  $\mathcal{S}(w)$  be the ball obtained by Theorem 1 when we used  $\tilde{w}$  as the suboptimal solution. Then,  $\{\tilde{w}_t\}_{t \in \mathbb{N}}$  satisfy the following property:*

$$\text{Vol}(\mathcal{S}(\tilde{w}_{t+1}) \cap \mathcal{S}(\tilde{w}_t)) < \frac{1}{2} \text{Vol}(\mathcal{S}(\tilde{w}_t)) \quad \forall t \in \mathbb{N}, \quad (32)$$

*where  $\text{Vol}(\mathcal{S})$  indicates the volume of  $\mathcal{S}$ .*

*Proof of Lemma 6.* By Theorem 1, the center  $m_t$  and the radius  $r_t$  of the ball  $\mathcal{S}(\tilde{w}_t)$  are written as

$$\begin{aligned} m_t &= \frac{1}{2} \left( \tilde{w}_t - C \sum_{i \in [n]} \nabla \ell_i(\tilde{w}_t) \right) = \tilde{w}_{t+1}, \\ r_t &= \frac{1}{2} \left\| \tilde{w}_t + C \sum_{i \in [n]} \nabla \ell_i(\tilde{w}_t) \right\|. \end{aligned}$$

Then,  $\forall t \in \mathbb{N}$ ,

$$\|m_{t+1} - m_t\|^2 = \|\tilde{w}_{t+2} - \tilde{w}_{t+1}\|^2 = \left\| -\frac{1}{2} \left( \tilde{w}_{t+1} + C \sum_{i \in [n]} \nabla \ell_i(\tilde{w}_{t+1}) \right) \right\|^2 = r_{t+1}^2.$$

It indicates that the center  $m_t$  is on the hypersphere of  $\mathcal{S}(\tilde{w}_{t+1})$ , i.e., there exists a half space  $\mathcal{H}_t$  whose boundary is the tangent hyperplane of  $\mathcal{S}(\tilde{w}_{t+1})$  at  $m_t$ . Using  $\mathcal{H}_t$ , we can show that

$$\text{Vol}(\mathcal{S}(\tilde{w}_{t+1}) \cap \mathcal{S}(\tilde{w}_t)) < \text{Vol}(\mathcal{H}_t \cap \mathcal{S}(\tilde{w}_t)) = \frac{1}{2} \text{Vol}(\mathcal{S}(\tilde{w}_t)).$$

□

Note that Lemma 6 holds for any loss functions  $\{\ell_i\}_{i \in [n]}$ . Thus, once we construct a ball including  $w_C^*$  as in Theorem 1, we can reduce the volume of the closed convex domain  $\mathcal{S}$  without any additional information, and it enables us to obtain tighter bounds.